Exploiting Monolingual Data at Scale for Neural Machine Translation

Lijun Wu[1,*], **Yiren Wang**[2,*], **Yingce Xia**[3,†], **Tao Qin**[3], **Jianhuang Lai**[1], **Tie-Yan Liu**[3]
[1]School of Data and Computer Science, Sun Yat-sen University;
[2]University of Illinois at Urbana-Champaign;
[3]Microsoft Research Asia;
[1]{wulijun3@mail2, stsljh@mail}.sysu.edu.cn;
[2]yiren@illinois.edu;
[3]{Yingce.Xia, taoqin, tyliu}@microsoft.com

Abstract

While target-side monolingual data has been proven to be very useful to improve neural machine translation (briefly, NMT) through back translation, source-side monolingual data is not well investigated. In this work, we study how to use both the source-side and target-side monolingual data for NMT, and propose an effective strategy leveraging both of them. First, we generate synthetic bitext by translating monolingual data from the two domains into the other domain using the models pre-trained on genuine bitext. Next, a model is trained on a noised version of the concatenated synthetic bitext where each source sequence is randomly corrupted. Finally, the model is fine-tuned on the genuine bitext and a clean version of a subset of the synthetic bitext without adding any noise. Our approach achieves state-of-the-art results on WMT16, WMT17, WMT18 English↔German translations and WMT19 German→French translations, which demonstrate the effectiveness of our method. We also conduct a comprehensive study on how each part in the pipeline works.

1 Introduction

Neural machine translation (briefly, NMT) (Bahdanau et al., 2014; Gehring et al., 2017; Wu et al., 2016; Vaswani et al., 2017) is well-known for its outstanding performance (Hassan et al., 2018), which usually relies on large-scale bitext for training. However, high quality bitext is always limited and costly to collect. In contrast, there exists large amount of monolingual data which can be leveraged to augment the training corpus. How to effectively leverage monolingual data is an important research topic for NMT and there are plenty of works studying this problem (Gulcehre et al., 2015, 2017; Sennrich et al., 2016a; He et al., 2016; Zhang and Zong, 2016; Zhang et al., 2018; Cheng et al., 2016; Wang et al., 2018; Chinea-Rios et al., 2017; Wu et al., 2018).

Among them, one of the most cited approach is back translation (briefly, BT) (Sennrich et al., 2016a), which leverages the target-side monolingual data. Specifically, a target-to-source translation model (trained on the genuine bitext) is used to translate target-side monolingual sentences into the source domain to generate a set of synthetic bitext, which is then used together with the genuine bitext to train a source-to-target NMT model. While target-side monolingual data is well utilized by NMT through BT and its variants (Sennrich et al., 2016a; Edunov et al., 2018; Hassan et al., 2018; He et al., 2016), the investigation on source-side monolingual data is very limited. Only few attempts (Zhang and Zong, 2016; Ueffing et al., 2007) are made to explore source-side monolingual data, with a common high-level idea that a source-to-target translation model is trained to translate the source-side monolingual data into target domain, the resulted synthetic data is then used for further training. In this work, we study how to leverage both source-side and target-side monolingual data to boost the accuracy of NMT.

We propose a simple yet effective strategy to leverage two-side monolingual data for NMT, which consists of three steps:
(1) *Preparation*: We pretrain a source-to-target and a target-to-source NMT models on the genuine bitext, and use them to generate synthetic bitext by translating the monolingual data from the source/target domain to the other domain respectively.
(2) *Large-scale Noised training*: The source sentences of the synthetic parallel corpus (including both the genuine source-side monolingual sentences and the synthetic source sentences back-

[*]The first two authors contributed equally to this work. This work was conducted at Microsoft Research Asia.
[†]Corresponding author.

translated from the target domain) are first corrupted. We then train an NMT model on this noised dataset together with the genuine bitext. We find that this step benefits from a large amount of monolingual data. The NMT model obtained from noised training can be further improved in the next finetune step.
(3) *Clean training*: We randomly generate a subset of the clean synthetic bitext without adding any noise, and leverage them together with the genuine bitext to finetune the output model of step (2). This step only needs a small set of synthetic bitext.

We conduct a comprehensive study of our proposed method on WMT English↔German translation and German↔French translation and have the following empirical observations:

- Using both source-side and target-side monolingual data is better than using monolingual data from only one domain (see Section 5.1).
- Adding noise to large-scale synthetic bitext improves the accuracy of NMT (see Section 5.2 and Section 5.3).
- Clean training/tuning of the model obtained from noised training further improves its accuracy (see Section 5.4).
- Our method achieves state-of-the-art results on English↔German newstest 2016, 2017 and 2018 and German→French newstest 2019 (see Section 4.2).

2 Related Work

Our work is related to several important research directions of NMT, and we describe the previous relevant works in this section.

2.1 Neural Machine Translation

NMT adopts the sequence-to-sequence framework, which consists of an encoder and a decoder in the network architecture. The encoder and decoder are usually built upon deep neural networks, which can be recurrent neural network (Sutskever et al., 2014), convolutional neural network (Gehring et al., 2017) or simple self-attention based transformer network (Vaswani et al., 2017). The encoder encodes the source sentence into a continuous representation space, and the decoder will decodes the target sentence based on the encoder representations word-by-word. The objective of the NMT model training is to maximize the conditional probability of the target sentence given the source sequence. Different model architectures and modifications have been proposed to improve the training efficiency and NMT accuracy (Hassan et al., 2018; Luong et al., 2015; Gu et al., 2016).

Different from the view of model architecture, in this paper, we study the NMT training from the data aspect. Specifically, we study the effect of both source-side and target-side monolingual data at scale and investigate how to make the best utilization of the monolingual data.

2.2 Improving NMT by Monolingual Data

NMT heavily relies on large-scale parallel dataset for training. To augment the limited bilingual data, there are plenty of works attempt to leverage the monolingual data to help the training, which includes the language model fusion (Gulcehre et al., 2015), back translation (Sennrich et al., 2016a), dual learning (He et al., 2016; Wang et al., 2018, 2019) and self learning (Zhang and Zong, 2016).

Gulcehre et al. (2017) integrates the hidden states from the target language model into the NMT decoder to improve the accuracy. Sennrich et al. (2016a) propose the back translation (BT) approach to leverage the target-side monolingual data, which is simple and effective. BT requires training an additional target-to-source NMT model given the bilingual dataset, the model will be used to back translate the target-side monolingual data. The translation output and the target-side monolingual data then paired as synthetic parallel corpus to augment the original bilingual dataset in order to further train the source-to-target NMT model. Dual learning (He et al., 2016) extends the BT approach to train NMT systems in both translation directions. When jointly training the source-to-target and target-to-source NMT models, the two models can provide back translated data for each other direction and perform multi-rounds BT. This strategy is also successfully adopted to build the unsupervised translation system (Lample et al., 2017).

There exists few attempts working on using the source-side monolingual data. Zhang and Zong (2016) propose self learning approach to generate the synthetic data for the source-side monolingual data, which is a semi-supervised method. Wu et al. (2017) leverage the source-side monolingual data to train the NMT system by learning reward func-

tion in a reinforcement learning framework.

2.3 Study of the Back Translation

Since BT is widely acknowledged and effective to improve the NMT model, there has been several works investigating back translation from different views. Poncelas et al. (2018) study on how using the back translated data as a training corpus (separately usage or combined with bilingual data) affects the performance of an NMT model. Burlot and Yvon (2019) analyze the accuracy impact from the quality of the BT data, the alternative uses of BT data to give explanations of why BT works. Cotterell and Kreutzer (2018) provide an interpretation of back translation as approximate inference in a generative model of bitext and give a new algorithm. However, above studies are all based on the small-scale monolingual data, therefore it remains unclear in the large-scale setting.

Edunov et al. (2018) firstly provide an extensive analysis of the back translation at scale. They investigate the different synthetic data generation methods, and also compare the different combination of the synthetic data with bitext. They finally build a strong system with millions of target-side monolingual sentences.

3 Training Strategy

In this section, we give a detailed introduction of our training strategy.

Notations: Let X and Y denote two languages, and let $\mathcal{X}$ and $\mathcal{Y}$ denote two corresponding language sentence domains, which are the collection of all sentences. Let $\mathcal{B} = \{(x_i, y_i)\}_{i=1}^{N}$ denote the bilingual training pairs, where $x_i \in \mathcal{X}$, $y_i \in \mathcal{Y}$, N is the number of sentence pairs. Let $\mathcal{M}_x = \{x_j\}_{j=1}^{M_x}$ and $\mathcal{M}_y = \{y_j\}_{j=1}^{M_y}$ denote the collections of monolingual sentences, where M_x and M_y are the sizes of the two sets, $x_j \in \mathcal{X}$, $y_j \in \mathcal{Y}$. Our objective is to obtain a translation model $f : \mathcal{X} \mapsto \mathcal{Y}$, that can translate sequences from language X to language Y.

There are three steps of our training strategy:

Step-1: Preparation. We first train two translation models $f_b : \mathcal{X} \mapsto \mathcal{Y}$ and $g_b : \mathcal{Y} \mapsto \mathcal{X}$ on the given bilingual data $\mathcal{B}$ by minimizing the negative log-likelihood. After that, we build the following two synthetic datasets through the trained models:

$$\begin{aligned} \bar{\mathcal{B}}_s &= \{(x, f_b(x))|x \in \mathcal{M}_x\}, \\ \bar{\mathcal{B}}_t &= \{(g_b(y), y)|y \in \mathcal{M}_y\}, \end{aligned} \tag{1}$$

where $\bar{\mathcal{B}}_s$ and $\bar{\mathcal{B}}_t$ can be seen the forward-translation (i.e., self learning) of source-side monolingual data (Zhang and Zong, 2016) and back-translation of target-side monolingual data (Sennrich et al., 2016a). In practice, f_b and g_b will output one translation by either beam search or random sampling according to the model output distribution. Random sampling explores more possibility for the unknown generated sentence, which may benefit the data augmentation. In this work, we mainly adopt beam search to generate the sentences, but we will also compare different sequence generation methods in Section 5.2.

Step-2: Large-scale noised training. Inspired from the noised back translation (Edunov et al., 2018) and denoising auto-encoder (Vincent et al., 2008), we add noise to the source-side data of both $\bar{\mathcal{B}}_s$ and $\bar{\mathcal{B}}_t$ for training instead of directly using them to train models. We build two following noised versions of the two augmented datasets,

$$\begin{aligned} \bar{\mathcal{B}}_s^n &= \{(\sigma(x), y)|(x, y) \in \bar{\mathcal{B}}_s\}, \\ \bar{\mathcal{B}}_t^n &= \{(\sigma(x), y)|(x, y) \in \bar{\mathcal{B}}_t\}, \end{aligned} \tag{2}$$

where σ is the operator of adding noise. We follow Edunov et al. (2018) to design σ. Specifically,
(1) we randomly replace a word in the sentence to be a special token "<UNK>" (representing unknown words) with probability 0.1;
(2) we randomly drop the words in each position with probability 0.1;
(3) we randomly shuffle (swap) the words in the sentence, with constraint that the words will not be shuffled further than three positions distance.

We then train an NMT model f_n for X to Y translation on $\mathcal{B} \cup \bar{\mathcal{B}}_s^n \cup \bar{\mathcal{B}}_t^n$ by minimizing the negative log-likelihood. Compared with Edunov et al. (2018), we enlarge the noised data from $\bar{\mathcal{B}}_t^n$ to $\bar{\mathcal{B}}_s^n \cup \bar{\mathcal{B}}_t^n$, instead of using the target-side monolingual data only. The intuition behind noised training is to force the encoder to discover more robust features and thus improve the generalization ability (Vincent et al., 2008). Besides, the output model of noised training has great potential to be improved in further finetune. Adding noise is widely acknowledged in other NLP tasks, like unsupervised NMT (Lample et al., 2017), BERT pretaining (Devlin et al., 2019), etc. There are some

other ways to augment/add noise to the training data (Artetxe et al., 2017) and we leave the combination with those approaches as future work.

Step-3: Clean data tuning. After obtaining noised training model from step-2, we further finetune it on the clean version of the synthetic data without adding noise manually. For the efficiency, we can randomly subsample $\bar{\mathcal{B}}_s$ and $\bar{\mathcal{B}}_t$ to form the clean $\bar{\mathcal{B}}_s^s$ and $\bar{\mathcal{B}}_t^s$ dataset. Then we continue tuning f_n on the new datasets:

$$\min \sum_{(x,y)\in\mathcal{B}\cup\bar{\mathcal{B}}_s^s\cup\bar{\mathcal{B}}_t^s} -\log P(y|x;f), \qquad (3)$$

where f is initialized by f_n.

There are different ways to obtain $\bar{\mathcal{B}}_s^s$ and $\bar{\mathcal{B}}_t^s$, such as the subset of $\bar{\mathcal{B}}_s$ and $\bar{\mathcal{B}}_t$ we described here. A more effective way is to train another $\bar{f}_b$ for X to Y translation and another $\bar{g}_b$ for Y to X translation, and use them to build new synthetic data for step-3. In this way, more diverse samples are included and we are able to achieve better results. We will provide more discussions in Section 4.1 and study in Section 5.4.

4 Experiments

We verify the effectiveness of our proposed training strategy in this section. We conduct experiments on four different translation tasks. We also make a comprehensive study about the effect of the monolingual data usage in our approach from various aspects.

4.1 Experimental Setup

Datasets We carry out experiments on four different translation tasks from WMT19 competition[1], including En→De, De→En, De→Fr and Fr→De, where En, De and Fr are short for English, German and French.

For En↔De translation tasks, the bilingual data consists of two parts: (1) We concatenate "Europarl v9", "News Commentary v14", "Common Crawl corpus" and "Document-split Rapid corpus", remove the empty and duplicate lines and eventually get a clean dataset of news domain. (2) We also use the Paracrawl dataset to extend the bilingual corpus. Consider Paracrawl is noisy, we apply a series of filtration rules to this dataset and remove the low-quality sentences, including sentences with too many punctuation marks or invalid characters, and those with too many or too few words, etc. All the rules are available in the `preprocess.py` in the supplementary document. These two parts of data are then merged together to get the bilingual dataset. We eventually get a clean corpus with about $5M$ clean data and $18M$ Paracrawl data, which are denoted as `WMT` and `WMTPC` respectively for ease of reference. The monolingual data we use is from newscrawl released by WMT19. We combine the newscrawl data from year 2016 to 2018 for the English and German monolingual corpus. After filtering with similar rules applied in Paracrawl (`preprocess_mono.py` in the supplementary document), an additional step is that we further perform language detection (Lui and Baldwin, 2012) on each side monolingual data. Finally, we randomly select about 120M sentences for each of English and German language. We choose newstest2015 as the validation set and use newstest2016 to newstest2019 as test sets.

For De↔Fr translation data, we follow the same process as that used in En↔De. Eventually, the `WMT` and ParaCrawl data of De↔Fr contains about 2M and 4.8M sentence pairs respectively. As for the French monolingual data, we use all the available newscrawl from previous years due to its small size compared to En and De. After filtering, we keep 60M monolingual data. We use the validation and test set provided by WMT19 for De↔Fr translation.

All datasets are tokenized with Moses (Koehn et al., 2007) toolkit[2]. The vocabulary is built based on the Byte-Pair-Encoding (BPE) (Sennrich et al., 2016b) with 35K merge operations[3] for both En↔De and De↔Fr datasets. We learn the BPE vocabulary jointly on the source and target language sentences.

Models We choose the state-of-the-art Transformer network as our model structure, which consists of an encoder with 6 layers and a decoder with 6 layers. We use `transformer_big` configuration for all experiments: the dimension of word embedding and the inner feed-forward layer is 1024 and 4096 respectively. The parameters of

[1] http://www.statmt.org/wmt19/translation-task.html

[2] https://github.com/moses-smt/mosesdecoder/tree/master

[3] https://github.com/rsennrich/subword-nmt

source and target word embeddings, as well as the projection layer before softmax are shared. The number of attention heads is 16. The dropout is fixed as 0.2 due to validation performance. We conduct experiments on the fairseq (Ott et al., 2019) codebase[4].

Training The models are trained by the Adam optimizer (Kingma and Ba, 2014) with $\beta_1 = 0.9$, $\beta_2 = 0.98$. We use the default learning rate schedule used in Vaswani et al. (2017) with the initial learning rate 5×10^{-4}. Label smoothing (Szegedy et al., 2016) is adopted with value 0.1. The batch size is set to be 4096 tokens per GPU. We use 4 P40 GPUs for training and update the parameters every 16 minibatches. The pretraining and the noised training both take about one week, and the finetune process takes about one day training.

To effectively leverage monolingual data, as mentioned in Section 3, we use two different groups of models to generate synthetic data for noised training and finetuning. For noised training, the models used for translating monolingual data are trained on WMT; for finetuning, the models are trained on WMTPC. The intuition behind using different generation models is that each optimization stage can benefit from the new/unseen generated synthetic training data. Therefore, we adopt this way in our experiments. As for the data size, in noised training, $|\bar{\mathcal{B}}_s| = |\bar{\mathcal{B}}_t| = 60M$, and during finetuning, $|\bar{\mathcal{B}}_s^s| = |\bar{\mathcal{B}}_t^s| = 20M$.

Evaluation To evaluate the model performance, we use beam search generation with beam width 5 and without length penalty. The BLEU score is measured by the de-tokenized case-sensitive SacreBLEU (Post, 2018), which is widely adopted in the WMT competitions[5].

4.2 Results

We summarize the results of our training strategy for En↔De and De↔Fr translations in Table 1 and Table 2.

As can be seen from Table 1, the Paracrawl dataset improves the model accuracy by a large margin. Compared with the baseline of using WMT only, on average, Paracrawl brings more than 3.0 and 4.0 BLEU improvements to En→De and De→En tasks respectively, indicating the effectiveness of leveraging more data. After large-scale noised training, the accuracy of each task is further boosted. On average, we improve the BLEU scores of En→De and De→En by 1.6 and 2.2 points, demonstrating the effectiveness of noised training. At last, the output models are tuned on the clean synthetic data. This step further brings 1.7 and 1.5 points gain over the previous step. For En↔De translation tasks, we finally achieve 40.9, 32.9, 49.2 and 43.8 BLEU scores from newstest2016 to newstest2019; for De→En, the eventual BLEU scores of the four test sets end up at 47.5, 41.0, 49.5 and 41.9. Such strong improvements over the baseline reveal the great potential of our proposed strategy.

	De→Fr	Fr→De
WMT	31.2	26.1
WMTPC	34.2	28.6
+Noised Training	36.1	30.8
+Clean Tuning	**37.3**	**33.1**

Table 2: De-tokenized case-sensitive SacreBLEU on WMT De↔Fr newstest2019. "+" is conducted upon WMTPC dataset.

Prior to this work, the most common way to leverage monolingual data is BT (Sennrich et al., 2016a). We also compare our strategy to the vanilla BT, which consists of $20M$ synthetic data and WMTPC. The results are shown in the last row of Table 1. Without finetuning, our noised training strategy surpasses the BT by 0.92 and 0.55 points respectively. This shows that our strategy is more effective than standard BT.

In Table 2, we show the results of De↔Fr translation tasks. We have similar observation as that for En↔De translations. Specifically, the noised training can improve WMTPC by 1.9 and 2.2 points on De→Fr and Fr→De. When turning to clean tuning, we can obtain another 1.2 and 2.3 points improvement. The results on De↔Fr demonstrate that our strategy works across different languages.

Our method achieves state-of-the-art results on En↔De newstest2016, newstest2017 and newstest2018 and De→Fr newstest2019. We list several widely acknowledged systems on En→De translation in Table 3: MS-Marian (Junczys-Dowmunt, 2018), which is the champion of

[4] https://github.com/pytorch/fairseq

[5] https://github.com/mjpost/sacreBLEU. SacreBLEU signatures: BLEU+case.mixed+lang.$task +numrefs.1+smooth.exp+test.$SET+tok.13a+version.1.2.12, where $task are en-de, de-en, de-fr and fr-de; $SET are 16, 17, 18, 19.

Model	En→De					De→En				
	2016	**2017**	**2018**	**2019**	**Avg**	**2016**	**2017**	**2018**	**2019**	**Avg**
WMT	34.0	28.0	41.3	37.3	35.15	38.6	34.3	41.1	34.5	37.13
WMTPC	37.1	30.5	45.6	40.3	38.38	41.9	37.5	45.4	40.1	41.23
+*Noised Training*	39.3	32.0	47.5	41.2	40.00	46.1	39.8	47.7	40.2	43.45
+Clean Tuning	**40.9**	**32.9**	**49.2**	**43.8**	**41.70**	**47.5**	**41.0**	**49.5**	**41.9**	**44.98**
WMTPC+BT	38.7	31.8	46.0	39.8	39.08	45.8	39.8	47.2	38.6	42.90

Table 1: De-tokenized case-sensitive SacreBLEU on WMT En↔De newstest2016, newstest2017, newstest2018, newstest2019 and the average score. "Avg" means the average BLEU score. "+" is conducted upon WMTPC dataset.

Model (En→De)	**2016**	**2017**	**2018**
FAIR (ensemble)	38.0	32.8	46.1
MS-Marian (ensemble)	39.6	31.9	48.3
Ours (single)	**40.9**	**32.9**	**49.2**

Table 3: De-tokenized case-sensitive SacreBLEU on WMT En→De newstest2016, newstest2017 and newstest2018. MS-Marian and FAIR are ensemble results while ours are single-model results.

Model (De→En)	**2016**	**2017**	**2018**
UCAM (ensemble)	45.1	38.7	48.0
RWTH (ensemble)	46.0	39.9	48.4
Ours (single)	**47.5**	**41.0**	**49.5**

Table 4: De-tokenized case-sensitive SacreBLEU on WMT De→En newstest2016, newstest2017 and newstest2018. UCAM and RWTH are ensemble results while ours are single-model results.

WMT18 En→De competition, and FAIR's model (Edunov et al., 2018) which leverages a large amount of monolingual data. Note that results of MS-Marian and FAIR are from ensemble models, while ours are from a single model. We find that our single model successfully surpasses the previous best systems in all test sets. Especially, for newstest2016 and newstest2018, we achieve 1.0 BLEU score improvement over the MS-Marian and set new records for these tasks. We also list the WMT18 top-2 systems for De→En translation in Table 4: RWTH (Graça et al., 2018) and UCAM (Stahlberg et al., 2018) systems, which are both ensemble models. Similarly, our single model surpasses these ensemble systems by a large margin.

5 Analysis

In this section, we provide a comprehensive study on the effect of monolingual data from different aspects, including data combination ways, data scale and the data generation methods. The experiments are conducted on the En→De translation and evaluated on newstest2016, newstest2017 and newstest2018. In this section, without specific clarification, the training data of each setting contains WMTPC.

5.1 Source or Target Monolingual Data

We first investigate the effect about different combinations of monolingual data. Specifically, we compare three different ways to use monolingual data, including leveraging source-side monolingual data only (i.e., $\bar{\mathcal{B}}_s$), target-side monolingual data only (i.e., $\bar{\mathcal{B}}_t$), and the monolingual data from both two sides (i.e., $\bar{\mathcal{B}}_s$ and $\bar{\mathcal{B}}_t$).

We keep the number of total synthetic data to be same across the three settings, that is, (i) 120M $\bar{\mathcal{B}}_s$; (ii) 120M $\bar{\mathcal{B}}_t$, and (iii) the combination of 60M $\bar{\mathcal{B}}_s$ and 60M $\bar{\mathcal{B}}_t$. We conduct this study on the clean synthetic data without noise, in order to verify which kind of data is more helpful for boosting performances.

The results are shown in Figure 1. From the figure, we can clearly observe that the best configuration is the combination of $\bar{\mathcal{B}}_s$ and $\bar{\mathcal{B}}_t$. With the same data size, on average, the model trained with combined synthetic dataset outperforms those trained with $\bar{\mathcal{B}}_s$ only and $\bar{\mathcal{B}}_t$ only by 2.6 and 3.5 points respectively. In particular, on newstest2018, the advantage of the combined dataset is nearly 5 points compared to $\bar{\mathcal{B}}_t$, which is extremely significant. This result strongly supports us that our training strategy by leveraging both the source-side and target-side monolingual

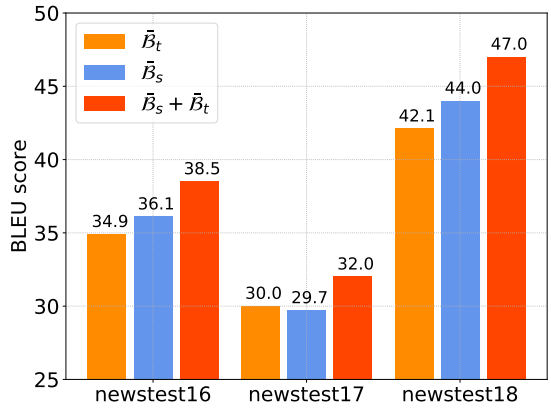


Figure 1: The de-tokenized SacreBLEU scores on En→De newstest2016, newstest2017 and newstest2018 of the models trained by different synthetic data: (1) $\bar{\mathcal{B}}_s$ from source-side monolingual data only, (2) $\bar{\mathcal{B}}_t$ from target-side monolingual data only and (3) the combination of $\bar{\mathcal{B}}_s$ and $\bar{\mathcal{B}}_t$.

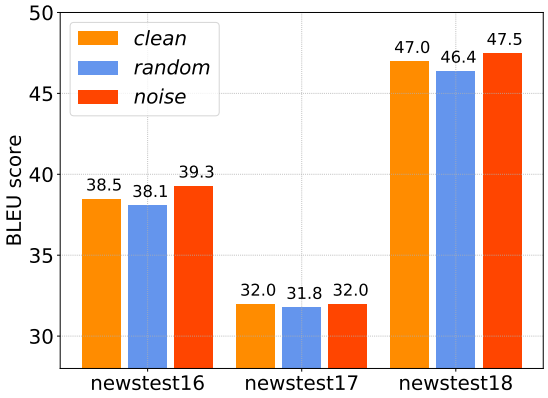


Figure 2: The de-tokenized SacreBLEU scores on En→De newstest2016, newstest2017 and newstest2018 of the models trained by synthetic data generated in different ways: (1) clean $\bar{\mathcal{B}}_s$ and $\bar{\mathcal{B}}_t$ data, (2) $\bar{\mathcal{B}}_s^r$ and randomly sampled $\bar{\mathcal{B}}_t^r$ data, and (3) noised $\bar{\mathcal{B}}_s^n$ and $\bar{\mathcal{B}}_t^n$ data.

data is helpful. Another point is that on such large-scale dataset, the $\bar{\mathcal{B}}_t$ data from back translation seems to be worse than the $\bar{\mathcal{B}}_s$ data generated by source-side monolingual data, which also supports the point that source-side monolingual is helpful.

5.2 Synthetic Data Generation

We conduct experiments on different generated synthetic data, to verify whether adding noise is essential. We use the same combined dataset (i.e., $60M$ $\bar{\mathcal{B}}_s$ and $60M$ $\bar{\mathcal{B}}_t$) as that used in Section 5.1 since the effectiveness has been verified.

We compare our noised training data $\bar{\mathcal{B}}_s^n$ and $\bar{\mathcal{B}}_t^n$ with another two baselines: 1) $\bar{\mathcal{B}}_s$ and $\bar{\mathcal{B}}_t$ without any transformation; 2) $\bar{\mathcal{B}}_s$ and the randomly sampled synthetic data $\bar{\mathcal{B}}_t^r$, where each token in the translation is sampled from the multinomial distribution determined by the NMT model. Random sampling is an alternative way to introduce noise, where the resulted synthetic dataset is more diverse but the translation quality is relatively poor. The data sizes for all settings are 120M.

We present the result in Figure 2. Overall, noised traininig outperforms the clean training, e.g., 0.8 and 0.5 points advantage on newstest2016 and newstest2018 respectively. Compared with randomly sampled data, our noised data training achieves more than 1.0 BLEU improvement on newstest2016 and newstest2018, and similar performance on newstest2017. The above results suggest that our noised training is effective.

5.3 Scale of Monolingual Data

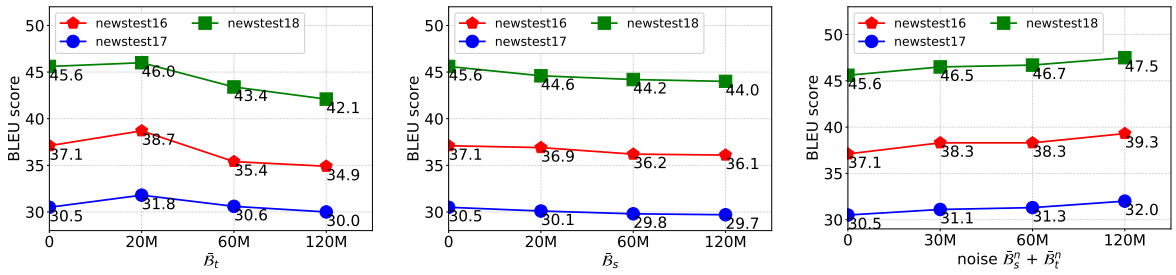
In this section, we give a comparison of different data scales for each kind of synthetic data.

(a) Scale of Target-side (BT) Data We first look into the widely adopted back translation data $\bar{\mathcal{B}}_t$. We vary the data scale from 20M to 60M, which are obtained by random selection from the 120M corpus, and then to 120M.

From Figure 3(a), we can observe that model performance is improved when we add 20M $\bar{\mathcal{B}}_t$ data. However, adding more clean back translated data starts to hurt the model performances. We suspect the reason is that the data distribution of $\bar{\mathcal{B}}_t$ shifts from the groundtruth distribution. Adding too much $\bar{\mathcal{B}}_t$ will enforce the model training to fit a biased distribution, making the generalization ability drop and thus, we observe that the BLEU scores drop. This result implies that we should choose a proper data size when using $\bar{\mathcal{B}}_T$ only, such as same size of bitext (e.g., 20M as we used).

(b) Scale of Source-side Data We then investigate the effect of data scale of source-side monolingual data, i.e., $\bar{\mathcal{B}}_s$ data. We also set the data sizes as {20M,60M,120M}.

The results are shown in Figure 3(b). We do not observe any improvement over the bilingual system when adding $\bar{\mathcal{B}}_s$ data, and with more data added, the performance slightly drops. It seems to be contrary to the experimental results in Zhang and Zong (2016) which are conducted on RNN models. We speculate the self learning approach



(a) Different scales of $\bar{\mathcal{B}}_t$ data. (b) Different scales of $\bar{\mathcal{B}}_s$ data. (c) Different scales of noised $\bar{\mathcal{B}}_s$ + $\bar{\mathcal{B}}_t$ data.

Figure 3: The de-tokenized SacreBLEU scores on newstest2016, newstest2017, newstest2018 of the models trained with varied data scales of (a) $\bar{\mathcal{B}}_t$ data, (b) $\bar{\mathcal{B}}_s$ data, and (c) combined $\bar{\mathcal{B}}_s$ and $\bar{\mathcal{B}}_t$ data.

is actually hard for the model to boost itself, since there are no other signals to help the learning. The results demonstrate that leveraging source-side monolingual data alone is not a good choice.

(c) Scale of Noised Synthetic Data We finally study our noised training on different sizes of noised synthetic data. The experiments are conducted on the combination of noised 30M, 60M and 120M $\bar{\mathcal{B}}_s^n$ and $\bar{\mathcal{B}}_t^n$ with the same number of source-side and target-side monolingual sentences[6]. We choose the maximum data size as 120M due to GPU memory limitation. The results are shown in Figure 3(c).

We can see that the performances are consistently improved as the number of the noised synthetic data increases on all test sets. The result again proves that the hybrid usage of source-side and target-side monolingual data is an effective approach which outperforms using the two kinds of data individually.

In summary, first, we verify that source-side monolingual data is helpful and the best way to use it is to combine with target-side monolingual data. Next, we show that adding noise to synthetic data outperforms that without noise. Finally, we empirically prove that our strategy benefits from more monolingual data, while BT does not, demonstrating our strategy has great potential of utilizing more data.

5.4 Synthetic Tuning

We further study the clean tuning step in our training strategy. Two questions remain to be answered: (1) Is it helpful to use two groups of models of building synthetic data for noised training and clean training as we discussed before? (2) Is it helpful to use noised training first regarding future BLEU score achieved by the finetune step?

[6]For example, same as combined 120M data, 60M combined data contains 30M $\bar{\mathcal{B}}_s^n$ and 30M $\bar{\mathcal{B}}_t^n$ data.

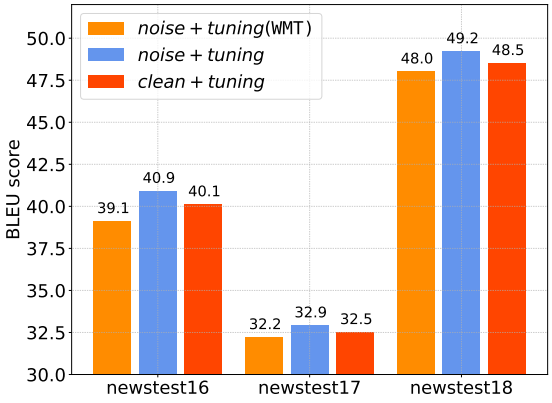


Figure 4: The de-tokenized SacreBLEU scores on En→De newstest2016, newstest2017 and newstest2018 of the models tuned by different synthetic data and pretrained models: (1) noised training model tuned on the subset of $\bar{\mathcal{B}}_s$ and $\bar{\mathcal{B}}_t$ (WMT in the figure), (2) noised training model tuned on the synthetic data as introduced in Section 4.1, and (3) clean training model tuned on the same synthetic data generated as (2).

To answer these questions, we conduct another two experiments: (1) At finetuning step, use a subsample of synthetic data of noised training step, which is $\bar{\mathcal{B}}_s^s \subset \bar{\mathcal{B}}_s, \bar{\mathcal{B}}_t^s \subset \bar{\mathcal{B}}_t$. (2) Training without noise first and then finetuning using clean data.

We present the results in Figure 4. Obviously, we can see that tuning on the subset of $\bar{\mathcal{B}}_s$ and $\bar{\mathcal{B}}_t$ is worse than tuning on another set of synthetic data used in our experiments (e.g., 49.2 v.s. 48.0 on newstest2018). In addition, first training on the clean synthetic data and then tuning on other clean synthetic data also makes improvement, but similar to the results shown in Section 5.2, it is still not as good as our noised pretraining. Above results

again prove the importance of our noised training and tuning on more diverse dataset is effective.

6 Conclusion

In this work, we exploit the monolingual data at scale for the neural machine translation. Different from previous works which usually adopt back translation on target-side monolingual data only, we propose an effective training strategy to boost the NMT performance by leveraging both source-side and target-side monolingual data. Our approach contains three steps: synthetic data generation, large-scale noised training on synthetic data and clean data training/tuning. We verify our approach on the widely acknowledged WMT English↔German translation tasks and achieve state-of-the-art results, as well as that for the WMT German↔French translations. We also make a comprehensive study on the monolingual data utilization.

For future work, we would like to verify our training strategy on more language pairs and other sequence-to-sequence tasks. Furthermore, we are interested in studying our noised training with other data augmentation approaches.

References

Mikel Artetxe, Gorka Labaka, Eneko Agirre, and Kyunghyun Cho. 2017. Unsupervised neural machine translation. *arXiv preprint arXiv:1710.11041*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Franck Burlot and François Yvon. 2019. Using monolingual data in neural machine translation: a systematic study. *arXiv preprint arXiv:1903.11437*.

Yong Cheng, Wei Xu, Zhongjun He, Wei He, Hua Wu, Maosong Sun, and Yang Liu. 2016. Semi-supervised learning for neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1965–1974.

Mara Chinea-Rios, Alvaro Peris, and Francisco Casacuberta. 2017. Adapting neural machine translation with parallel synthetic data. In *Proceedings of the Second Conference on Machine Translation*, pages 138–147.

Ryan Cotterell and Julia Kreutzer. 2018. Explaining and generalizing back-translation through wake-sleep. *arXiv preprint arXiv:1806.04402*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. *NAACL*.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 489–500.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. 2017. Convolutional sequence to sequence learning. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1243–1252. JMLR. org.

Miguel Graça, Yunsu Kim, Julian Schamper, Jiahui Geng, and Hermann Ney. 2018. The rwth aachen university english-german and german-english unsupervised neural machine translation systems for wmt 2018. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 377–385.

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor OK Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1631–1640.

Caglar Gulcehre, Orhan Firat, Kelvin Xu, Kyunghyun Cho, Loic Barrault, Huei-Chi Lin, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2015. On using monolingual corpora in neural machine translation. *arXiv preprint arXiv:1503.03535*.

Caglar Gulcehre, Orhan Firat, Kelvin Xu, Kyunghyun Cho, and Yoshua Bengio. 2017. On integrating a language model into neural machine translation. *Computer Speech & Language*, 45:137–148.

Hany Hassan, Anthony Aue, Chang Chen, Vishal Chowdhary, Jonathan Clark, Christian Federmann, Xuedong Huang, Marcin Junczys-Dowmunt, William Lewis, Mu Li, et al. 2018. Achieving human parity on automatic chinese to english news translation. *arXiv preprint arXiv:1803.05567*.

Di He, Yingce Xia, Tao Qin, Liwei Wang, Nenghai Yu, Tie-Yan Liu, and Wei-Ying Ma. 2016. Dual learning for machine translation. In *Advances in Neural Information Processing Systems*, pages 820–828.

Marcin Junczys-Dowmunt. 2018. Microsoft's submission to the wmt2018 news translation task: How i learned to stop worrying and love the data. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 425–430.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th annual meeting of the association for computational linguistics companion volume proceedings of the demo and poster sessions*, pages 177–180.

Guillaume Lample, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2017. Unsupervised machine translation using monolingual corpora only. *arXiv preprint arXiv:1711.00043*.

Marco Lui and Timothy Baldwin. 2012. langid. py: An off-the-shelf language identification tool. In *Proceedings of the ACL 2012 system demonstrations*, pages 25–30. Association for Computational Linguistics.

Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*.

Alberto Poncelas, Dimitar Shterionov, Andy Way, Gideon Maillette de Buy Wenniger, and Peyman Passban. 2018. Investigating backtranslation in neural machine translation. In *Proceedings of the 21st Annual Conference of the European Association for Machine Translation: 28-30 May 2018, Universitat d'Alacant, Alacant, Spain*, pages 249–258. European Association for Machine Translation.

Matt Post. 2018. A call for clarity in reporting bleu scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Improving neural machine translation models with monolingual data. In *ACL*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1715–1725.

Felix Stahlberg, Adrià de Gispert, and Bill Byrne. 2018. The university of cambridges machine translation systems for wmt18. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 504–512.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826.

Nicola Ueffing, Gholamreza Haffari, and Anoop Sarkar. 2007. Transductive learning for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 25–32.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. 2008. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103. ACM.

Yijun Wang, Yingce Xia, Li Zhao, Jiang Bian, Tao Qin, Guiquan Liu, and Tie-Yan Liu. 2018. Dual transfer learning for neural machine translation with marginal distribution regularization. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Yiren Wang, Yingce Xia, Tianyu He, Fei Tian, Tao Qin, ChengXiang Zhai, and Tie-Yan Liu. 2019. Multi-agent dual learning. In *International Conference on Learning Representations*.

Lijun Wu, Fei Tian, Tao Qin, Jianhuang Lai, and Tie-Yan Liu. 2018. A study of reinforcement learning for neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3612–3621.

Lijun Wu, Li Zhao, Tao Qin, Jianhuang Lai, and Tie-Yan Liu. 2017. Sequence prediction with unlabeled data by reward function learning. In *IJCAI*, pages 3098–3104.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Jiajun Zhang and Chengqing Zong. 2016. Exploiting source-side monolingual data in neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1535–1545.

Zhirui Zhang, Shujie Liu, Mu Li, Ming Zhou, and Enhong Chen. 2018. Joint training for neural machine translation models with monolingual data. In *Thirty-Second AAAI Conference on Artificial Intelligence*.